

How To... Integrate the TinyMCE JavaScript Content Editor in Web Page Composer

Applicable Releases:

Portal for SAP NetWeaver 7.3

Version 1.0

April 2011



HOW-TO GUIDE

© Copyright 2011 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice.

These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Document History

Document Version	Description
1.00	First official release of this guide

Typographic Conventions

Type Style	Description
<i>Example Text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
Example text	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Icons





Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

Table of Contents

1. Business Scenario	2
2. Background Information	2
3. Prerequisites.....	2
4. TinyMCE JavaScript Editor Component.....	3
4.1 Expected Result.....	3
4.2 Scope.....	3
4.2.1 TinyMCE.java	3
4.2.2 TinyMCEComponent.java	3
4.2.3 TinyMCEComponentRenderer.java.....	4
4.2.4 TinyMCERFServiceWrapper.java.....	4
4.2.5 ITinyMCERFServiceWrapper.java.....	4
4.2.6 PortalApp.xml	4
4.2.7 Scripts	5
5. Integration into Web Page Composer.....	6
5.1 Deploy the EAR file.....	6
5.2 Work through the system configuration steps	6
5.3 Update the XML file	6
5.4 Restart the portal	7
6. Result.....	8
7. Appendix.....	9

1. Business Scenario

The TinyMCE editor provides a feature-rich alternative to the HTML editor that comes standard with the Web Page Composer. Using the Web Page Composer editor API, you can replace the standard editor with the TinyMCE editor.

This How-to Guide explains the basic steps you need to perform to replace the standard editor, including the required development and the configuration changes needed to integrate the new component.

2. Background Information

Make sure that the following components have been successfully installed:

- SAP NetWeaver Developer Studio (corresponding version for your SAP NetWeaver Portal)
- SAP NetWeaver 7.3 including Portal and Web Page Composer

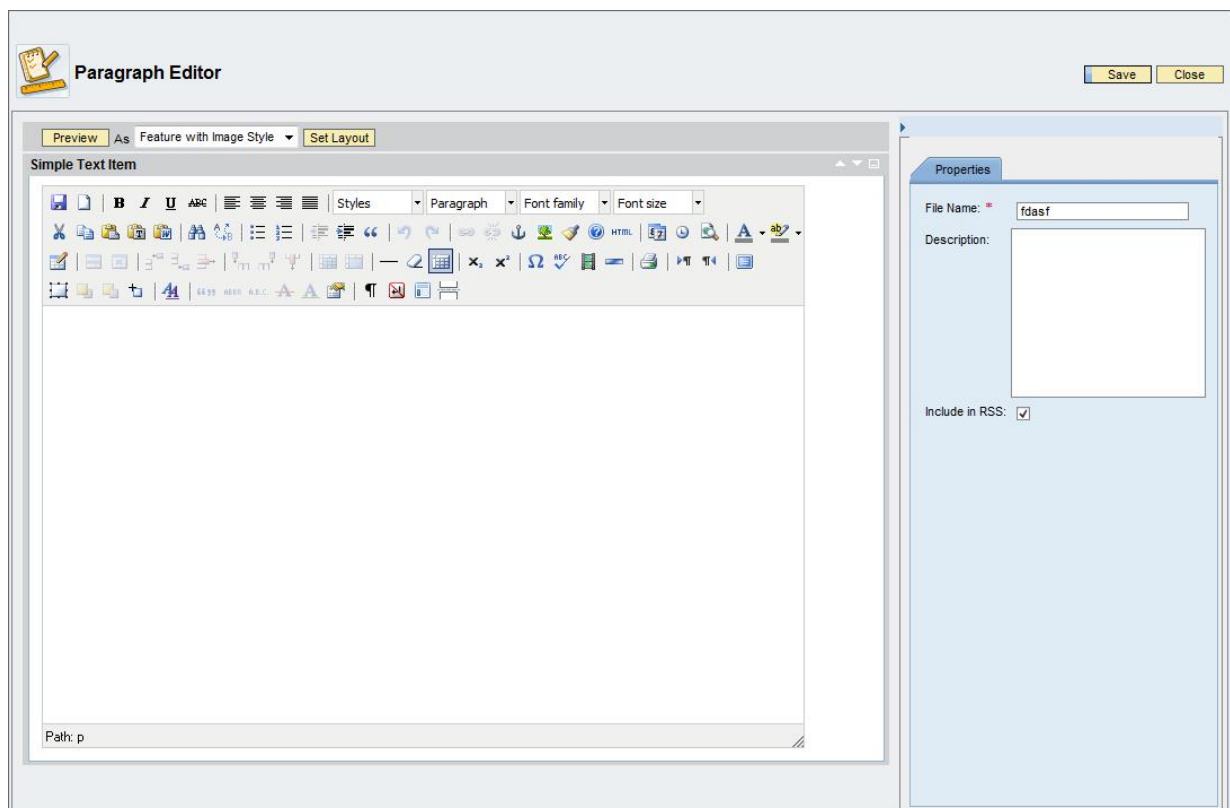
3. Prerequisites

- The *System Admin* and *Content Admin* portal roles are assigned to your user.
- The *WPC Editor* role is assigned to your user (required for testing purposes only).

4. TinyMCE JavaScript Editor Component

4.1 Expected Result

The goal of our basic integration example is to replace the SAP default HTML editor, which is used for editing paragraphs. After you have completed the coding and configuration tasks, you should be able to display the new editor when creating or editing a paragraph in Web Page Composer.



4.2 Scope

This document outlines the core ideas behind the Web Page Composer editor API and provides an introduction to the basic functions of the core classes and methods. This How-to Guide aims at providing additional information and guidance for understanding the Web Page Composer editor API. It is not intended to be a complete documentation set.

4.2.1 TinyMCE.java

The `TinyMCE.java` class implements the `Component` interface. It is the wrapper that represents the editor. The initialization string for the editor is generated here.

4.2.2 TinyMCEComponent.java

The `TinyMCEComponent.java` class implements the `AbstractEditorComponent` interface. The following methods have to be overridden:

- **public** Element getXMLElement(Document document, String elementId, String tagName, **boolean** includePropertyValues) {}
This method creates an XML element to be added to the output XML file.
- **public void** initializeFromPageContext(AbstractEditorObject abstractEditor, IPageContext context, ComponentConfig cc) {}
This method initializes the composite editor from the page context. The localization takes place in this method as well.
- **public void** initialiseFromXML(String xmlContent, Attributes attributes, IPropertyMap properties, AbstractEditorObject object, IPortalComponentRequest request) {}
This method performs the same actions as initializeFromPageContext() but initialization data is taken from the XML file.
- **private** Component getTinyMCE(TinyMCE tiny, AbstractEditorObject abstractEditor) {}
This method creates a new TinyMCE object.
- **public void** init(AbstractEditorObject editorObject, ComponentConfig cc) {}
This method calls the initializeFromPageContext() method.
- **public** Element getDefaultXMLElement(Document document, String elementId, String tagName, String defaultValue, IPropertyName propName) {}
This method creates a default XML element to be added to the output XML file.
- **public** Component getHtmlbComponent() {}
This method gets the HTMLB Component.

4.2.3 TinyMCEComponentRenderer.java

The `TinyMCEComponentRenderer.java` class implements the `HtmlbRenderer` interface. This class renders the editor for the output on the screen.

4.2.4 TinyMCERFServiceWrapper.java

The `TinyMCERFServiceWrapper.java` class implements the `ITinyMCERFServiceWrapper` interface. In this class the registration of the deployable unit is handled.

4.2.5 ITinyMCERFServiceWrapper.java

The `ITinyMCERFServiceWrapper.java` class extends the `IService.java` class. Every deployable unit should have a unique key for the purpose of identification and class loading. This key is defined here.

4.2.6 PortalApp.xml

The `portalapp.xml` file is the *deployment descriptor* for all portal applications. In the case of our Editor project, this file describes all the application and service configurations.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<application>
```



```
<application-config>
  <property name="SharingReference" value="com.sap.nw.wpc.core,
com.sap.nw.wpc.core, com.sap.nw.wpc.km.svc.extresourceservice,
knowledgemanagement, landscape, htmlb, exportalJCOclient, exportal,
com.sap.netweaver.kmc.people" />
</application-config>
<components/>
<services>
  <service name="TinyMCERFServiceWrapper">
    <service-config>
      <property name="className"
value="com.sample.km.wpc.editor.TinyMCERFServiceWrapper" />
      <property name="startup" value="true" />
    </service-config>
    <service-profile/>
  </service>
</services>
</application>
```

4.2.7 Scripts

The following folder contains all JavaScript files for the TinyMCE Editor.
You can [download the files from SCN](#)

5. Integration into Web Page Composer

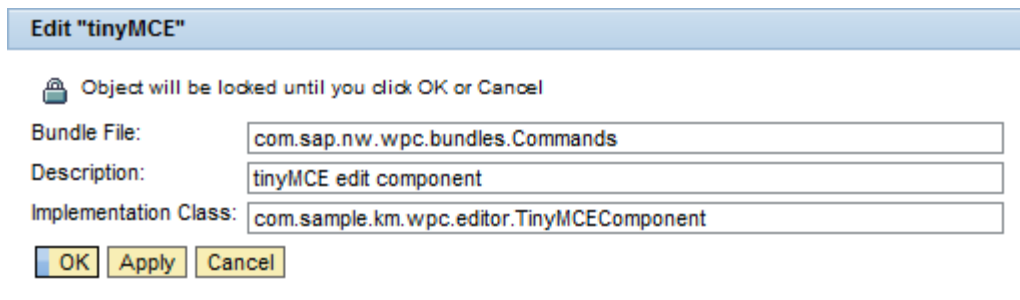
This chapter describes the steps that you need to perform in order to integrate the TinyMCE editor into Web Page Composer.

5.1 Deploy the EAR File


1. Import the attached ZIP file into your SAP NetWeaver Developer Studio.
2. Export the project as an EAR file.
3. Deploy the EAR file to the portal.

5.2 Work Through the System Configuration Steps

1. In the portal, navigate to *System Administration* → *System Configuration* → *Knowledge Management* → *Content Management* → *Web Page Composer* → *Editor* → *Editor Components* → *Components*.
2. Choose *tinyMCE*, and choose *Edit*.
3. Make sure that the following entries are inserted:



Edit "tinyMCE"

 Object will be locked until you click OK or Cancel

Bundle File:

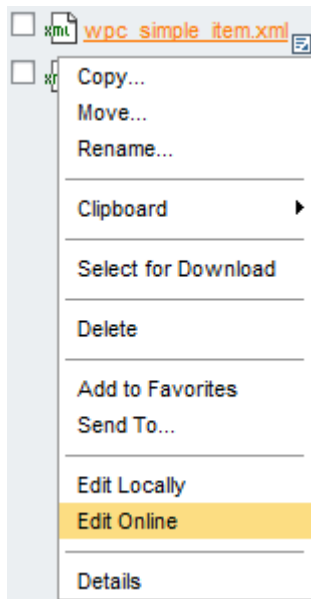
Description:

Implementation Class:

5.3 Update the XML File

To switch from the default HTML editor to the TinyMCE editor, you have to edit the XML file which describes the elements.

1. In the portal navigate to *Content Administration* → *KM Content* → *etc* → *wpceditor* → *types*.
2. Open the context menu of *wpc_simple_item.xml* and choose *Edit Online*.



3. Replace

```
<elements>  
  <element id="text" description="xml.xlbl.text" type="htmledit"  
default="true" singleinstance="true" nodelete="true"/>  
</elements>
```

with

```
<elements>  
  <element id="text" description="xml.xlbl.text" type="tinyMCE"  
default="true" singleinstance="true" nodelete="true"/>  
</elements>
```

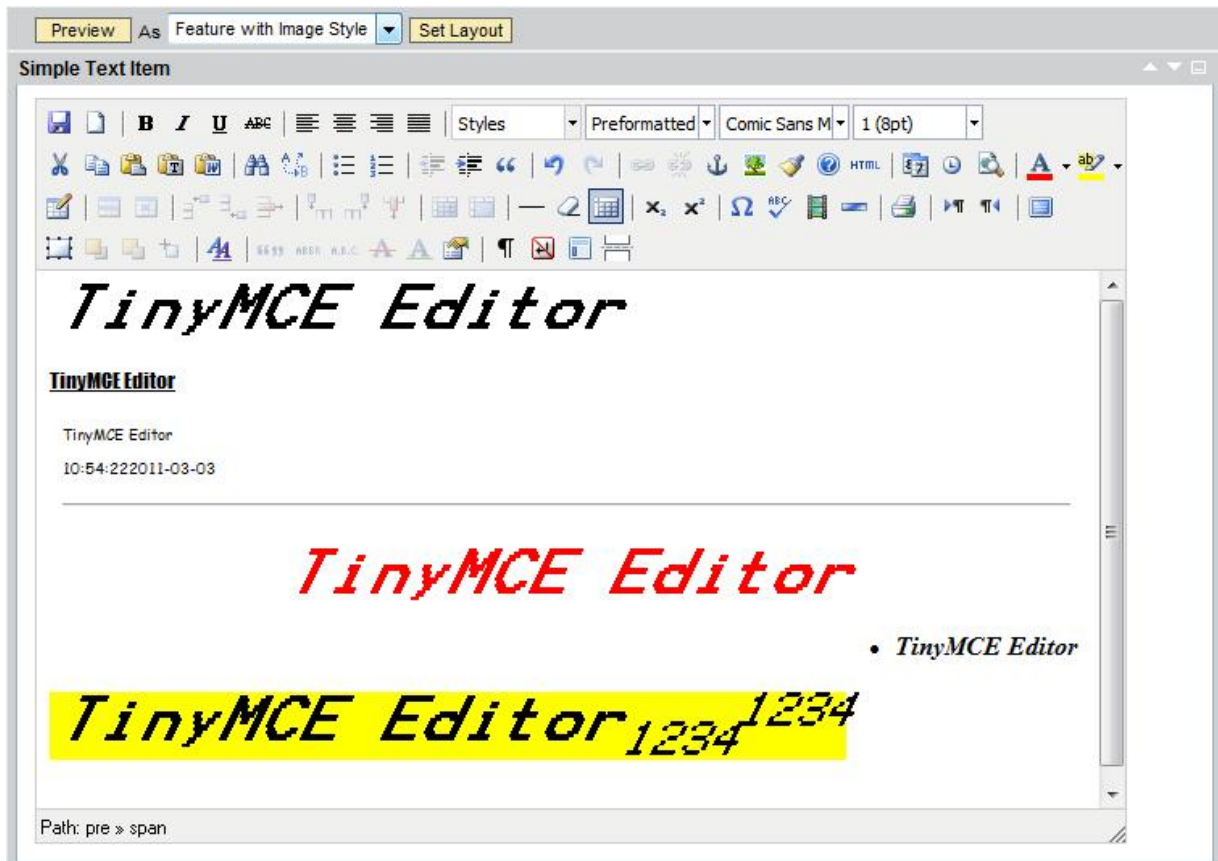
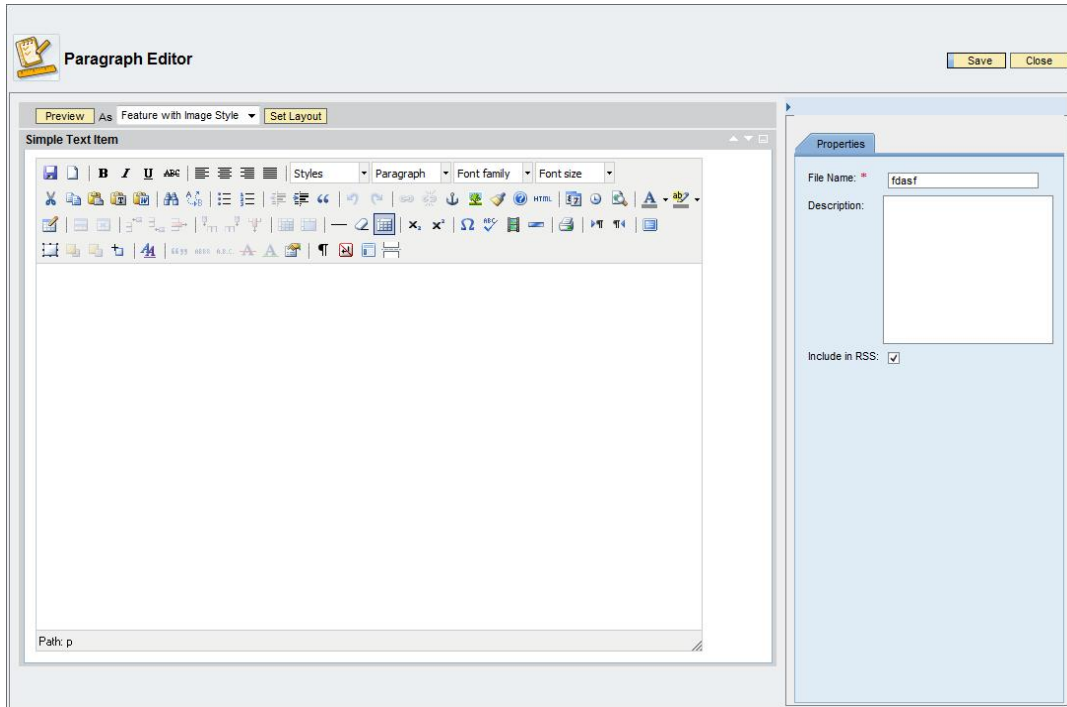
4. Save your changes..

5.4 Restart the Portal

After you have performed the deployment and configuration steps, you must restart the portal. Otherwise your changes will not take effect.

6. Result

After completing the necessary steps you can create paragraphs with the TinyMCE editor.



7. Appendix

Appendix A - Abbreviations

Abbreviation	Description
WYSIWYG	“What You See Is What You Get”
WPC	Web Page Composer
API	Application Programming Interface
NWDS	SAP NetWeaver Developer Studio

www.sdn.sap.com/irj/sdn/howtoguides